Alumno: Roberto Gregorio Moreno
Ingeniería de Telecomunicación

Universidad de Paderborn (Alemania)
Tutor: M. Sc. Carlos Paiz Gatica
Cotutor UC3M: Mario Muñoz Organero

RESUMEN PFC:
Modelo de un péndulo invertido basado en FPGA para simulaciones Hardware-in-the-Loop en tiempo real

EXTRACTO

Este proyecto fin de carrera consiste en un estudio de la dinámica de un péndulo invertido especifico, en este caso, el péndulo invertido utilizado por el departamento System and Circuit Technolgy (SCT) del Heinz Nixdorf Institute en la universidad de Paderborn (Alemania). El objetivo del estudio dinámico es crear un modelo realista de este péndulo invertido, que, posteriormente, será implementado en Hardware usando la tecnología FPGA (Field-Programmable Gate Array). El flujo de diseño utilizado en este modelo esta basado en Matlab Simulink y en Xilinx System Generator. El modelo en Hardware obtenido será un buen sustituto para el péndulo real y, por tanto, será una herramienta perfecta para testear los controladores usando simulaciones Hardware-in-the-Loop en tiempo real.

1. INTRODUCCIÓN

Este proyecto fin de carrera se basa en el péndulo invertido usado en el departamento System and Circuit Technolgy (SCT) del Heinz Nixdorf Institute en la universidad de Paderborn (Alemania). Este péndulo es un banco de pruebas para algoritmos de control. El objetivo es mantener en equilibrio la plomada del péndulo cuando esta está por encima de su eje, es por esto que se llama péndulo invertido. Esta posición *invertida* es claramente inestable, pero usando un sistema de realimentación y unos dispositivos llamados controladores es posible mantener en equilibrio el péndulo.

El péndulo invertido no tiene sentido desde el punto de vista de producto comercial, pero se usa en muchos departamentos de electrónica porque es un sistema simple y con una dinámica conocida que permite testear fácilmente las técnicas de control, las cuales son los verdaderos productos comerciales. Algunas aplicaciones comerciales que utilizan estos dispositivos controladores son: el guiado de cohetes, haciendo que estos sigan la trayectoria planeada; el auto-balance de los transportes personales (e.g. Segway PT), haciendo que este siempre esté en equilibrio; o el guiado de las grúas utilizadas en los astilleros, donde la precisión y la estabilidad en sus movimientos es fundamental.

En concreto, en el departamento de SCT el péndulo invertido sirve para testear el uso de hardware re-configurable en el dispositivo controlador, el cual esta implementado en hardware usando la tecnología FPGA. Estas FPGAs están integradas en una tarjeta llamada RAPTOR, la cual es un prototipo modular rápido basado en FPGA y desarrollado por el departamento de SCT.

El sistema de banco de pruebas de los controladores consiste en conectar las señales de salida del péndulo invertido a la tarjeta RAPTOR a través de la interfaz de su bus PCI. Después estas señales son enviadas (utilizando el bus local) al modulo concreto de la tarjeta RAPTOR donde se encuentran los controladores implementados en una memoria FPGA del tipo Virtex II Pro. Allí, los controladores calculan el próximo *set point* para el péndulo y envían una señal con el voltaje correspondiente para alcanzar ese punto deseado. De esta forma es como se cierra el bucle de realimentación que hace posible que el péndulo llegue y se mantenga en la posición invertida.

Profundizando un poco más en el concepto de hardware re-configurable éste consiste en el rápido intercambio de la configuración inicial de una memoria FPGA a otra configuración distinta (es decir un circuito diferente) en un momento determinado. Las ventajas de usar el hardware re-configurable en vez de usar dos FPGAs, o el uso de FPGAs en vez de microcontroladores, (velocidad, bajo coste, etc.) que son explicadas con más profundidad en la literatura referenciada en la documentación original.

El hardware re-configurable lo utilizamos en nuestro caso concreto porque tenemos dos circuitos diferentes, correspondientes a los dos controladores, pero una sola FPGA para implementarlos. Los circuitos controladores son: el controlador de balanceo, el cual se encarga de llevar al péndulo la posición invertida (180º) columpiándolo; y el controlador de equilibrado, el cual se encarga de mantener el péndulo en equilibrio en esa posición invertida. Inicialmente la FPGA de los controladores esta configurada para el controlador de balanceo, y cuando el péndulo cumple ciertas condiciones (posición muy cercana a la invertida, velocidad angular baja, etc.) se produce el cambio de circuito controlador y se implementa el de equilibrado. Este cambio debe ser lo suficientemente rápido como para que el péndulo no se caiga, es decir, tiene que funcionar bien en tiempo real.

Otro punto importante y que debe ser tenido en cuenta sobre los controladores es que el diseño de estos comienza a partir de un modelo realista del sistema a controlar (también conocido como planta). En nuestro caso, los controladores utilizados en este proyecto fin de carrera fueron diseñados a partir de un modelo muy básico e inexacto del péndulo, por tanto, tras la obtención de nuestro nuevo modelo mucho más realista, estos controladores pueden ser mejorados.

Finalmente queremos introducir el concepto de realidad aumentada, una tecnología que proporciona una visión directa o indirecta donde se mezclan elementos reales y virtuales (modelados en 3D). Este concepto es muy interesante para nosotros porque el modelo hardware del péndulo que obtendremos como resultado al final de este proyecto fin de carrera, estará implementado en un chip (FPGA). Por tanto, utilizando la realidad aumentada podemos sustituir las aburridas graficas que utilizamos para mostrar las salidas del péndulo, con las que corroboramos que los controladores funcionan correctamente, por una *mágica* imagen de un péndulo virtual en 3D que se moverá en tiempo real de la misma forma que lo haga nuestro modelo hardware obtenido.

Por tanto, la realidad aumentada será la pareja perfecta para nuestro modelo del péndulo invertido implementado en una FPGA, puesto que los dos juntos serán un sustituto preciso y virtual del péndulo invertido real utilizado en el departamento de SCT.

Tras esta breve introducción podemos mostrar el propósito y los objetivos de este proyecto fin de carrera realizado en la universidad de Paderborn (Alemania).

1.1. Propósito

Encontrar un modelo realista del péndulo utilizado en el departamento de SCT e implementarlo en hardware usando una FPGA, la cual se encuentra en un modulo (diferente al de los controladores) de la tarjeta RAPTOR. La meta es poder sustituir al péndulo real por este modelo consiguiendo, de esta manera, una herramienta ideal (pequeña y manejable) para probar los controladores en tiempo real y mediante simulaciones del tipo Hardware-in-the-Loop. Y finalmente, comprobar la compatibilidad de este modelo hardware con la tecnología de realidad aumentada en tiempo real.

1.2. Objetivos

Existen múltiples beneficios para el departamento de SCT como resultado de la realización de este proyecto fin de carrera, por ejemplo:

- El modelo en hardware podrá remplazar al péndulo real, que pesa más de 30 Kg. y ocupa muchísimo espacio, haciendo que el sistema, utilizado como banco de pruebas para controladores en tiempo real, sea mucho más pequeño y manejable. Por tanto, estos tests que antes solo se podían realizar en el laboratorio del departamento de SCT, tras este proyecto fin de carrera, se podrán realizar prácticamente en cualquier sitio, incluso en casa, puesto que solamente será necesario un ordenador y una tarjeta RAPTOR.

- Los tests en tiempo real serán más rápidos, baratos y seguros que cuando se utiliza el péndulo real porque no existen elementos reales como la etapa de potencia o el motor AC, que requieren tiempo (encendido o inicialización), energía (el motor necesita corriente de alta intensidad) y medidas de seguridad (necesarias para evitar riesgos como la alta corriente o los peligrosos movimientos del péndulo).

- Se podrán realizar test extremos (que pueden dañar el sistema real) sin problemas porque el modelo en hardware nunca romperá su motor, su correa de transmisión o su husillo, por ejemplo.

- Este nuevo modelo mucho más exacto y preciso del sistema del péndulo invertido podrá utilizarse para mejorar los diseños de los circuitos controladores. Recordemos que todos los diseños de controladores se basan en las ecuaciones que representan a la dinámica del sistema a controlar, y tras este proyecto fin de carrera las ecuaciones obtenidas serán mucho más precisas que las anteriores, y, por tanto, los controladores serán mejores.

2. MODELADO DEL PÉNDULO

Este capítulo comienza con una breve descripción del sistema, sus elementos y características, incluyendo una breve explicación del comportamiento del sistema del péndulo invertido utilizado por el departamento de SCT. En el diagrama mostrado en la figura II.1 podemos observar, de una manera gráfica, el sistema a modelar.

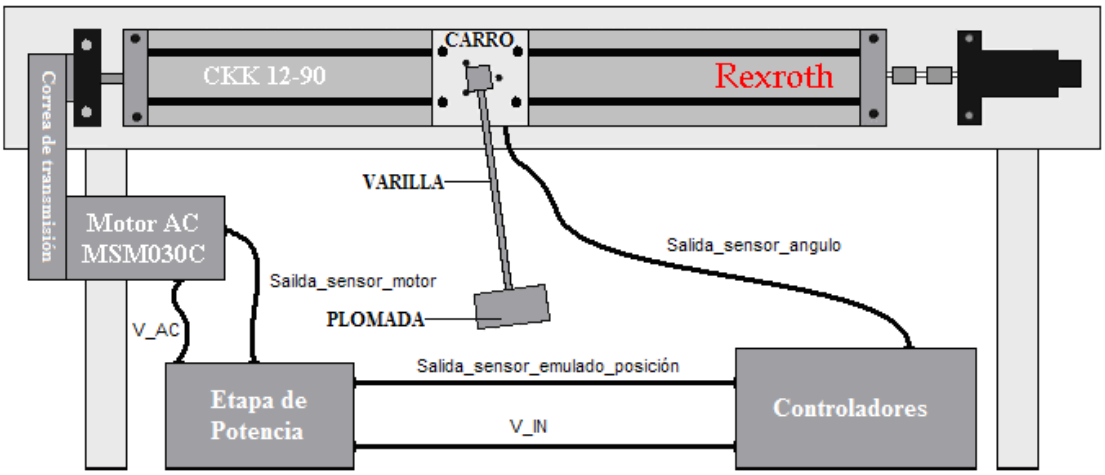

Figura II. 1: Diagrama del sistema de péndulo invertido del departamento de SCT

A continuación comenzamos el modelado del sistema, y lo primero que hacemos es subdividirlo en bloques para facilitar la ardua tarea del modelado. En la figura II.2 observamos el diagrama completo de bloques resultante, y, sobre un fondo gris, podemos ver los boques del sistema del péndulo invertido que serán modelados en este proyecto fin de carrera (es decir, todos los bloques excepto el de los controladores).

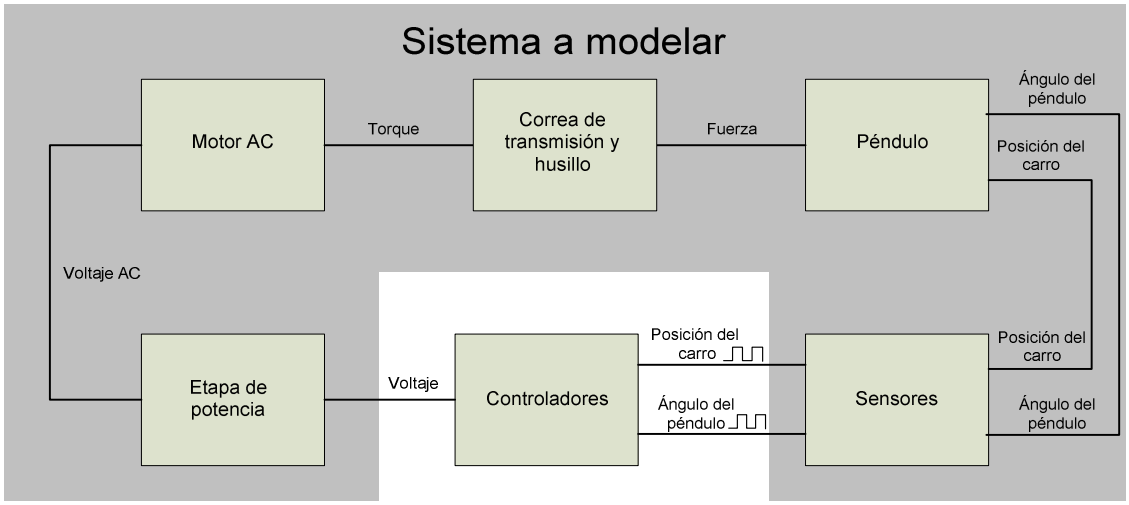

Figura II.2: Diagrama de bloques del sistema a modelar (péndulo invertido departamento SCT)

En este esquema de cinco bloques a modelar se puede hacer una simplificación puesto que existen tres bloques que corresponden a la parte mecánica del sistema (etapa de potencia, motor AC y correa de transmisión y husillo) que pueden agruparse en uno solo.

Por tanto, finalmente tenemos tres partes a modelar: la primera corresponde al bloque mecánico del sistema, el cuál tiene como entrada la señal de voltaje suministrada por el bloque de los controladores y como salida una fuerza que moverá el péndulo; la segunda parte importante es el bloque del péndulo, el cuál será modelado utilizando ecuaciones diferenciales y aplicando un modelo preciso de fricción; y, por último, la parte que corresponde al bloque de los sensores, el cuál convierte las salidas del bloque del péndulo en señales TTL, que, posteriormente, serán introducidas como entrada en la el bloque de los controladores. Estas señales son fundamentales para que los controladores calculen el siguiente set point, el cuál será suministrado mediante una señal de voltaje al bloque mecánico, cerrando, de esta forma, el lazo de realimentación.

El bloque mecánico fue modelado utilizando un modelo lineal en el que la fuerza de salida es directamente proporcional al voltaje de entrada.

El bloque del péndulo fue modelado utilizando el método analítico, y, más concretamente, usamos el mecánica lagrangiana para describir las ecuaciones diferenciales que rigen el comportamiento del péndulo. Incluyendo en este modelo a las casi siempre despreciadas (por la literatura consultada) fuerzas de rozamiento estático y dinámico, debido a que buscábamos un modelo muy exacto del péndulo y despreciar estas fuerzas introduciría pequeñas diferencias con la realidad.

El bloque de los sensores (o codificadores rotatorios relativos) fue modelado mediante una maquina finita de estados, que en función de la posición angular del eje a sensar daba como salida un 1 ó 0, generando, de esta forma, el tren de pulsos que requieren como entrada los controladores.

Finalmente este capítulo finaliza uniendo los modelos del bloque mecánico y del péndulo para generar un modelo no lineal de caja gris. Este nombre un poco atípico, se debe a que las ecuaciones del péndulo son no lineales y a que algunos parámetros del modelo son conocidos (caja blanca) y otros desconocidos (caja negra), por tanto, la nomenclatura utilizada es la de caja gris.

Por último, terminamos dando valores numéricos a cada uno de los parámetros del modelo, ya sea por medio de medidas experimentales (véase la masa del péndulo), de cálculos (por ejemplo el momento de inercia), o mediante la identificación de parámetros usando una herramienta de Matlab, para la cuál es necesaria la realización previa de algunos experimentos con el péndulo real.

Sin duda la parte más complicada de este proyecto fin de carrera fue la identificación de parámetros del modelo no lineal de caja gris. Debido a la alta complejidad que conlleva estimar los parámetros de un modelo de caja gris cuando la dinámica del sistema que representa (en nuestro caso, el péndulo invertido utilizado por el departamento de SCT) se comporta de manera no lineal.

3. REPRESENTACIÓN DEL MODELO EN SIMULINK

Este capítulo se centra en encontrar un circuito equivalente al modelo hallado en el capítulo anterior utilizando el software de Matlab llamado Simulink. El cuál es un entorno de programación virtual que funciona sobre el entorno Matlab y que posee un nivel de abstracción más alto que éste. Con este software podemos crear un esquemático en el que implementaremos nuestro modelo del péndulo utilizando los bloques Simulink. Estos bloques, que aparecen ordenados en librerías, encapsulan diferentes funciones de Matlab, ya sean operaciones matemáticas sencillas (sumas, restas, multiplicaciones, divisiones, etc.), más complicadas (exponenciales, integrales, etc.), de algebra de Boole (and, or, not, etc.) o funciones de conexión de señales (multiplexores, conmutadores, etc.).

Por tanto, el punto más importante tratado en este capítulo es la conversión de las ecuaciones diferenciales no lineales que rigen nuestro modelo en un circuito con bloques Simulink.

Para ello usamos bloques integradores, en vez de los bloques derivadores, que en principio parecen ser la equivalencia obvia. Pero la solución no es tan trivial, pues los bloques derivadores no pueden almacenar condiciones iniciales y, por tanto, el péndulo siempre empezaría desde la posición inicial (es decir, el péndulo abajo). Pero, con el sistema real se puede empezar con el péndulo arriba, por ejemplo, para hacer una prueba aislada del controlador de equilibrado. Por tanto, este requisito (incluir condiciones iniciales) es fundamental en nuestro modelo del péndulo. Además los bloques integradores suelen dar mejores resultados que los derivadores.

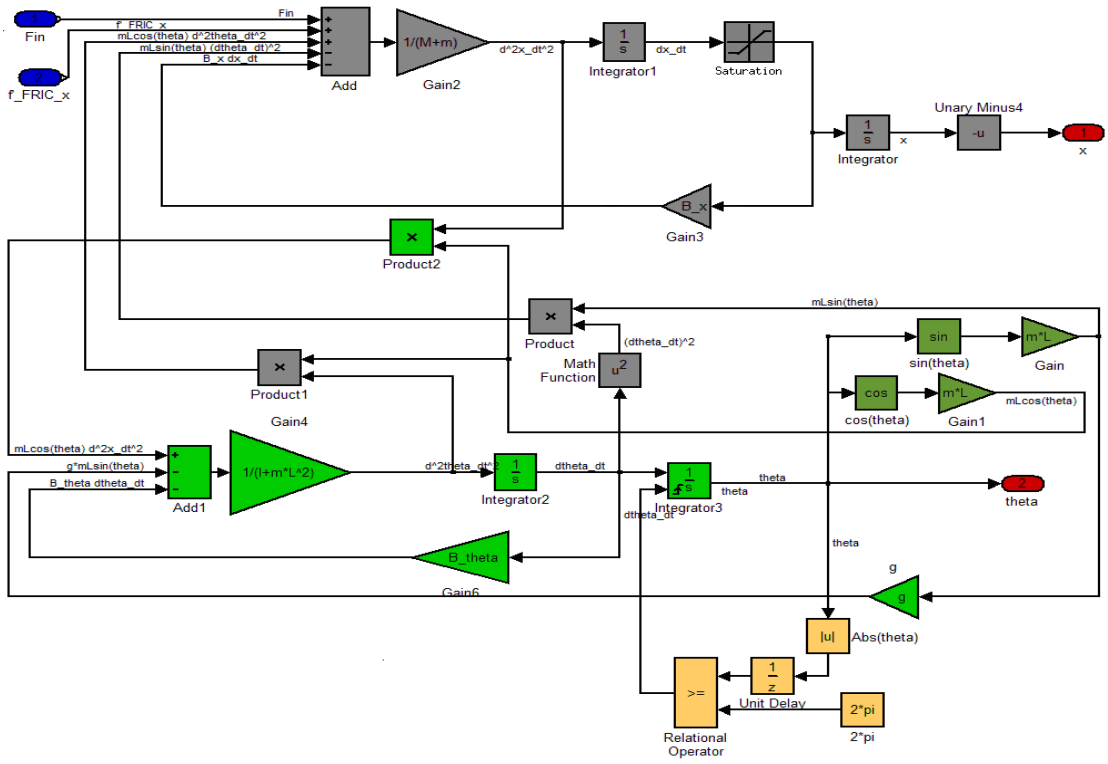

Figura III. 1: Parte del modelo del péndulo en Simulink

A partir de estos bloques integradores fue relativamente sencillo hallar el modelo en Simulink que representara las ecuaciones del sistema obtenidas en el capítulo anterior. A modo de ejemplo podemos observar en la figura III.1 una parte del modelo del péndulo en Simulink.

Este capítulo también trata la realización en Simulink de nuestro modelo de sensores. Prácticamente los dos esquemáticos son iguales, aunque dependiendo de si se trata del sensor de la posición del carro o de la posición angular del péndulo, existen unas mínimas diferencias que están explicadas con detalle en la memoria original, como, por ejemplo, la diferente resolución de los dos sensores.

Conviene destacar que Simulink suministra unos bloques específicos para la maquinas finitas de estados y que obviamente estos fueron los elegidos para nuestro modelo de sensores. Este bloque puede apreciarse en la figura III.2 en verde oliva (denotado como *Moore Chart*) donde se muestra el esquemático de Simulink para el sensor de posición del carro.

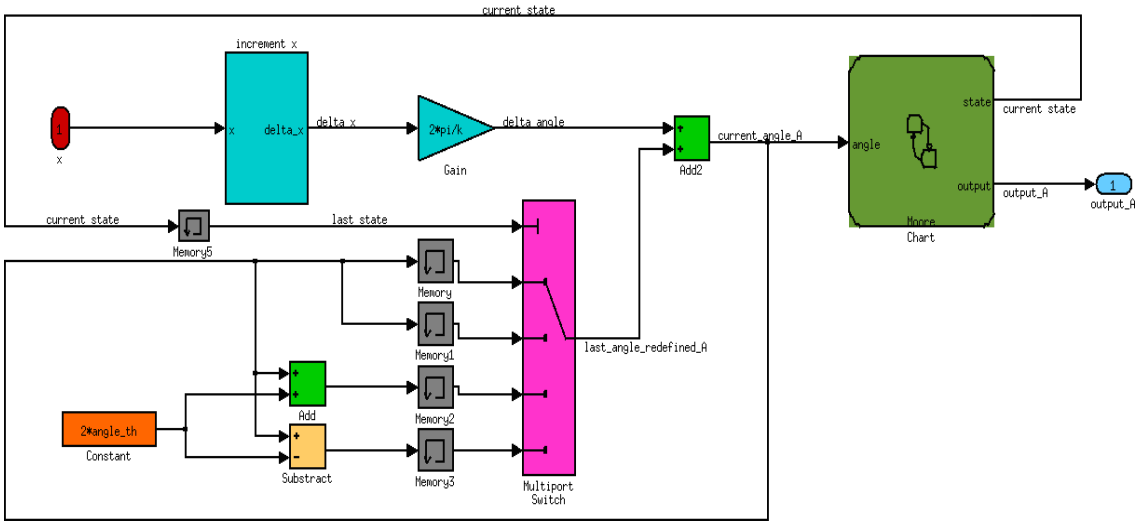

Figura III.2: Esquemático de Simulink para el sensor de posición del carro (canal A)

La funcionalidad de estos bloques Simulink específicos que representan a una maquina finita de estados viene dada mediante una herramienta suministrada por Matlab a tal efecto que se llama StateFlow, y que, de una forma gráfica sencilla e intuitiva, permite al usuario describir el diagrama de flujo con el que trabajará esa máquina finita de estados.

Por último el capítulo finaliza con una comparación entre nuestro modelo en Simulink del péndulo con el real. Para ello se utilizaron datos tomados de un experimento con el sistema real, donde los controladores consiguen estabilizar el péndulo en la posición invertida. También se probó el correcto funcionamiento de nuestro modelo de sensores en Simulink tomando unos datos reales de la posición del carro y de la posición angular del péndulo, pasándolos por el sensor donde se codificó esta información, y usando el bloque decodificador, que utilizan los controladores, para obtener el valor numérico transmitido por los sensores mediante un tren de pulsos. Si este valor numérico coincide con el dato real inicial el sensor funciona correctamente.

4. DESCRIPCIÓN EN HARDWARE DEL MODELO

En este capítulo el modelo en software del péndulo hallado en el capitulo anterior (usando bloques de Simulink) es convertido en un modelo en hardware, que, finalmente, será implementado en una FPGA alojada en el módulo derecho la tarjeta RAPTOR mientras que los controladores se encuentran en el módulo izquierdo (ver figura IV.1 donde se aprecia la configuración final del sistema).

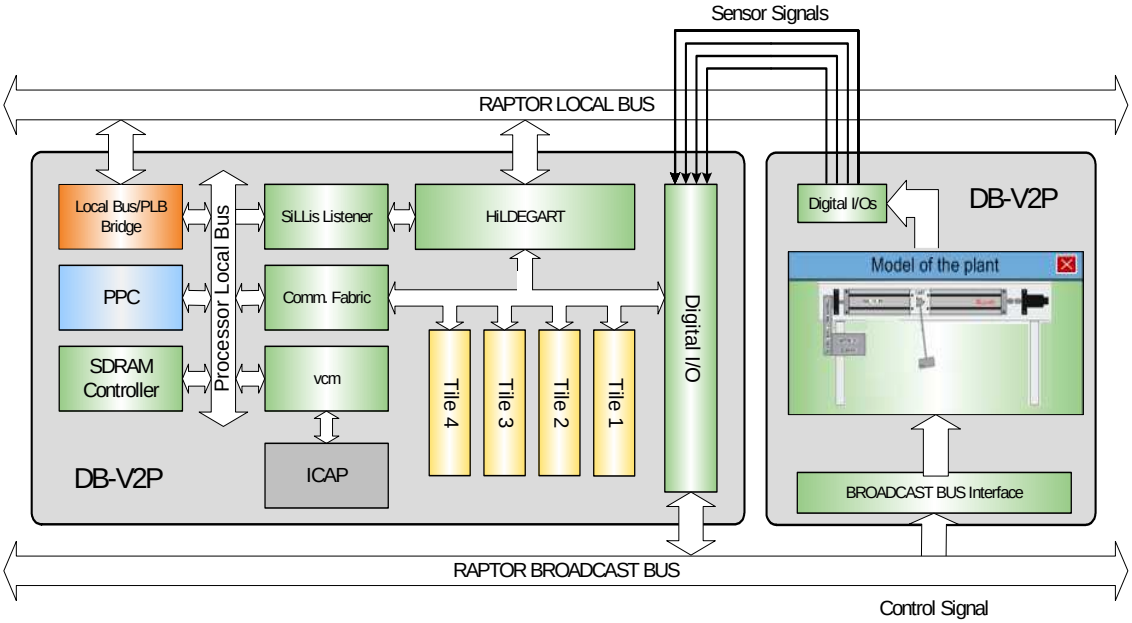

Figura IV.1: Diagrama de la configuración final de la tarjeta RAPTOR con los controladores (modulo izquierdo) y con el modelo hardware del péndulo (módulo derecho)

Por tanto, la primera sección del capítulo consiste en remplazar los bloques Simulink por otros bloques que están orientados a la futura implementación en hardware y que están suministrados por la empresa de la FPGA: Xilinx. Estos nuevos bloques se encuentran en una librería especial que Xilinx diseño para Simulink llamada System Generator.

En teoría la conversión del modelo software al modelo hardware es un paso muy sencillo, pero lamentablemente no todos los bloques de Simulink tienen una equivalencia directa en los bloques de System Generator. Por tanto, en esta parte de la memoria se muestran las soluciones ingenieriles adoptadas para encontrar la equivalencia entre bloques deseada.

Otro de los problemas solucionados en la conversión del modelo software del péndulo invertido al modelo hardware fue la adaptación al sistema de notación de punto fijo utilizado por Xilinx en sus esquemáticos creados con bloques de System Generator. Por tanto, debimos transformar nuestro modelo Simulink con notación en coma flotante con base decimal y formato *long*, en una notación en punto fijo con base binaria y con complemento a dos para los números negativos.

Tras obtener el esquemático final del modelo del péndulo invertido realizado por completo con bloques de System Generator, pasamos a generar su correspondiente lista de nodos mediante la compilación de este esquemático. Esta lista de nodos será convertida en un fichero VHDL mediante un software desarrollado por el departamento de SCT llamado V-MAGIC. Este fichero VHDL resultante contendrá unas líneas de programación extra necesarias para la futura compatibilidad con la interfaz de la tarjeta RAPTOR.

Finalmente, creamos el bitstream mediante el software suministrado por Xilinx llamado ISE Project Navigator. Este bitstream resultante representa a nuestro modelo del péndulo y con él será programada la FPGA Virtex II Pro del módulo derecho de la tarjeta RAPTOR (recordar figura IV.1).

Por último en este capítulo se realizan algunas pruebas donde: se compara este modelo en hardware con el modelo en software y el péndulo real; probamos al modelo en hardware del sistema de péndulo invertido del departamento de SCT en una simulación off-line (no en tiempo real) de FPGA-in-the-Loop, donde el modelo del péndulo se encuentra implementado en la FPGA de la tarjeta RAPTOR que le corresponde pero el bloque de los controladores se encuentra en el entorno virtual de Simulink; y por último se prueba al modelo en hardware del péndulo en una simulación on-line (en tiempo real) de FPGA-in-the-Loop, donde tanto el modelo del péndulo como los controladores se encuentran implementados en sus correspondientes FPGAs dentro de la tarjeta RAPTOR, tal y como se muestra en el diagrama de la figura IV.1, y usando la realidad aumentada (ver figura IV.2) para reemplazar los aburridos gráficos de salida del péndulo (x, posición del carro, y θ, posición angular del péndulo) por una imagen virtual del péndulo.

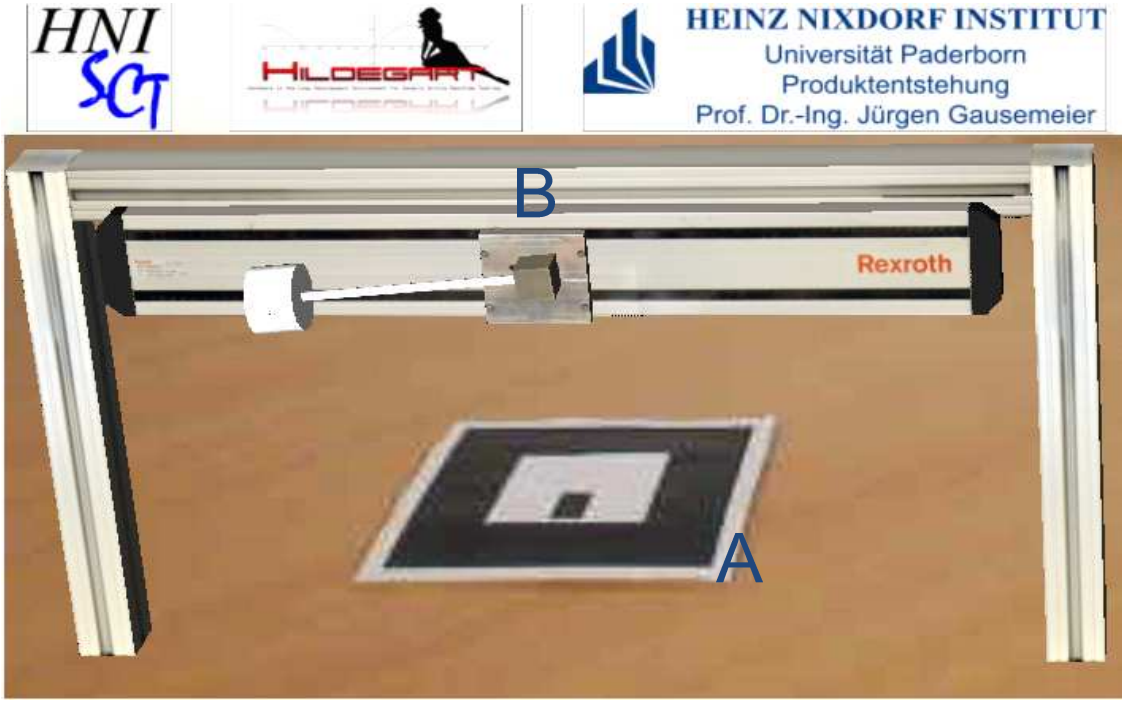

Figura IV.2: Imagen tomada de la pantalla del ordenador que muestra la imagen virtual del péndulo moviendose (B) y la cartulina con su patrón asociado (A)

5. CONCLUSIONES

El modelo final basado en FPGA del péndulo invertido usado por el departamento de SCT no es exacto al real, es un poco diferente, pero la mayoría de los objetivos propuestos al comienzo de este proyecto fin de carrera han sido alcanzados:

Este modelo es una herramienta ideal para testar a los controladores de manera separada mediante simulaciones FPGA-in-the-Loop en tiempo real o no. Decimos ideal porque gracias a este modelo no es necesario usar el sistema real y, por tanto, es mucho más cómodo, pequeño, barato y seguro que el banco de pruebas real.

Por supuesto que los test más extremos, que podían dañar al sistema real y en consecuencia anteriormente no se podían hacer, ahora se pueden realizar sin problemas porque nuestro modelo nunca romperá el motor o la correa de transmisión.

También este modelo puede resultar muy útil al departamento de SCT porque pude ser utilizado para diseñar nuevos controladores que sean mejores que los actuales. Recordad que los controladores actuales están basados en un modelo de péndulo estándar y menos exacto (por no considerar las fuerzas de fricción estática y dinámica) que él hallado por nosotros.

La compatibilidad del modelo basado en FPGA con la realidad aumentada ha sido demostrada, y, de hecho, podríamos decir que forman una pareja perfecta, puesto que los dos juntos son un sustituto virtual del péndulo invertido real.

Por otro lado en este capítulo final del proyecto fin de carrera remarcamos algunos resultados parciales obtenidos que son muy interesantes y que podrían aplicarse en futuros proyectos del departamento de SCT. Por ejemplo, el modelo de los sensores, que tanto en software como en hardware, da unos magníficos resultados (errores entre 10^{-4} y 10^{-5}) y que han sido modelados de una manera original y orientada a hardware nunca vista en la literatura consultada. Otro ejemplo de estos resultados parciales que pueden ser utilizados en un futuro por el departamento de SCT son las equivalencias en hardware (con bloques de System Generator) de los bloques Simulink que no tenían equivalencia directa en la librería suministrada por Xilinx, véase bloques integradores, valor absoluto, multiplicador por signo o bloque de saturación.

Y finalmente destacamos que las posibles futuras mejoras que se realicen a este proyecto deben estar centradas en las pequeñas diferencias entre nuestro modelo y el sistema real para conseguir un sustituto exacto y virtual del péndulo invertido utilizado en el departamento de SCT. Sabemos que la mayoría de estas diferencias son debidas a la dificultad de caracterizar el péndulo invertido mediante la identificación de parámetros de un modelo no lineal de caja gris y que pueden ser solucionadas realizando un mejor proceso de identificación de parámetros.

Gracias a la estructura de nuestro modelo esos futuros cambios pueden ser fácilmente incluidos en nuestro proyecto, simplemente deberemos poner los nuevos valores de los parámetros (hallados en el futuro y mejorado proceso de identificación) dentro del script de Matlab donde se definen las constantes de nuestro modelo y no hará falta cambiar nada más. Como podemos ver es una estructura totalmente reutilizable.